Authorization for Grid Computing

International Symposium on Grid Computing

Taipei,Taiwan

Mar 10, 2003

Mary Thompson

Lawrence Berkeley National Laboratory

Acknowledgements to Marty Humphrey

University of Virginia

Global Grid Forum

Security Services

- **Integrity**: Maintaining data consistency
- **Confidentiality**: Protection from disclosure to unauthorized persons
- **Authentication**: Assurance of identity of person or originator of data
- **Non-repudiation**: Originator of communications can't deny it later
- **Authorization**: Rights to perform some action
- **Availability**: Resources available for authorized parties

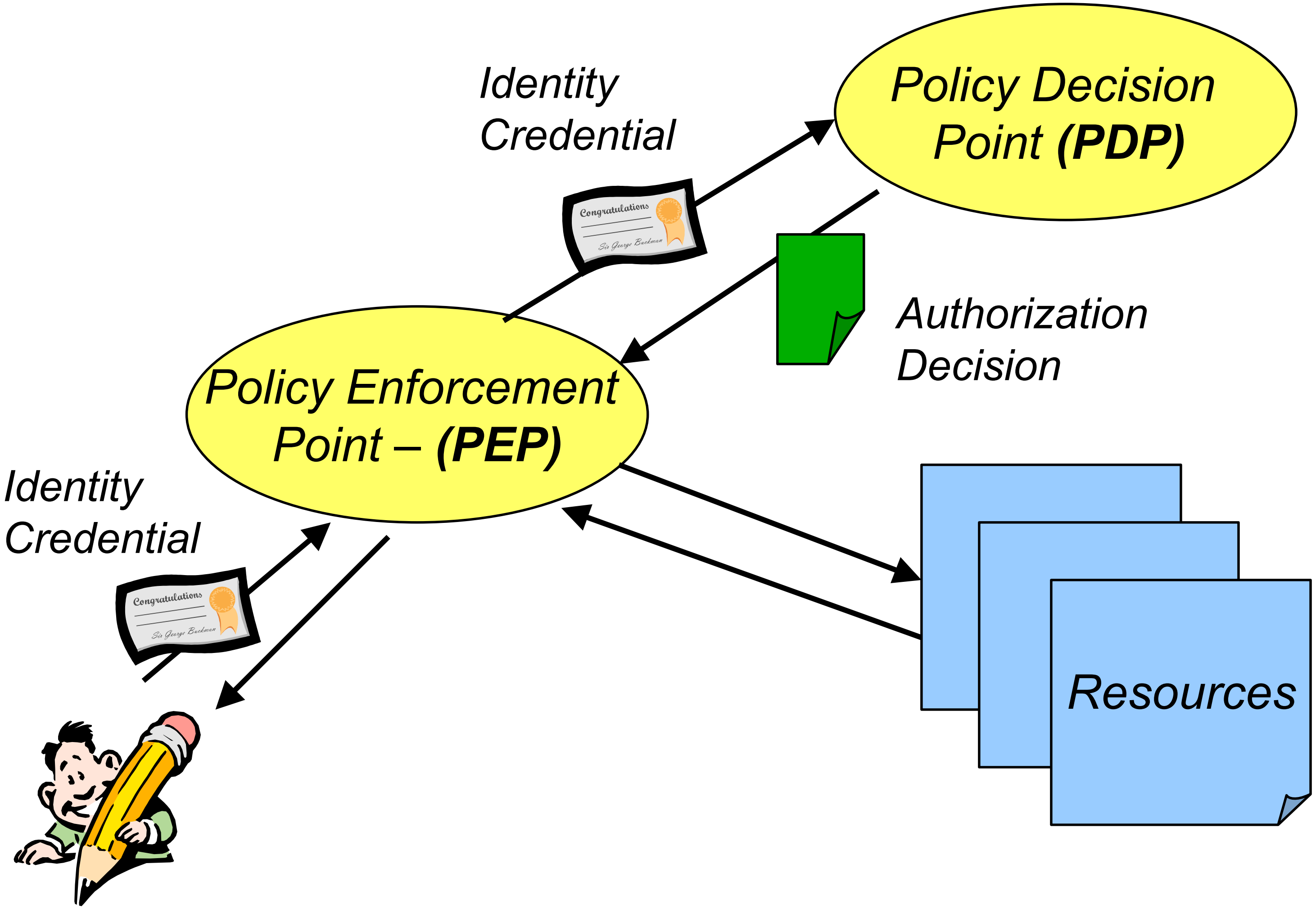
Authorization Models

- **Identity based (pull model)**
  - Authenticated user identity is presented to the resource
  - The resource gatekeeper evaluates access policy or calls an authorization service to decide the user's rights
  - Most common – Unix, AFS, DCE
- **Token based (push model)**
  - A token aka capability (unforgeable ticket) grants the holder to rights to a resource
  - Used in academic operating systems .e.g. Hydra, Chorus circa 1975-80
  - Delegation is easy (maybe too easy); Revocation and unforgeability requirement have been the major drawbacks

Authorization Models (cont)

- **Trusted Third Party**
  - The user provides his identity and the handle of a trusted authorization agent to call.
  - If the resource gateway trusts the agent, it is called with the user's identity token and returns the authorization decision.
  - Web services are using this now
    - Passport , Shibboleth, Liberty Alliance
- **Agent based**
  - User makes request to agent for resources, and agent authorizes the request
  - Agent calls resource gateway and "provisions' the resources for the user (network bandwidth reservations)
  - User uses the resources from the resource gateway

Pull Model



Identity Credential

Policy Decision Point ***(PDP)***

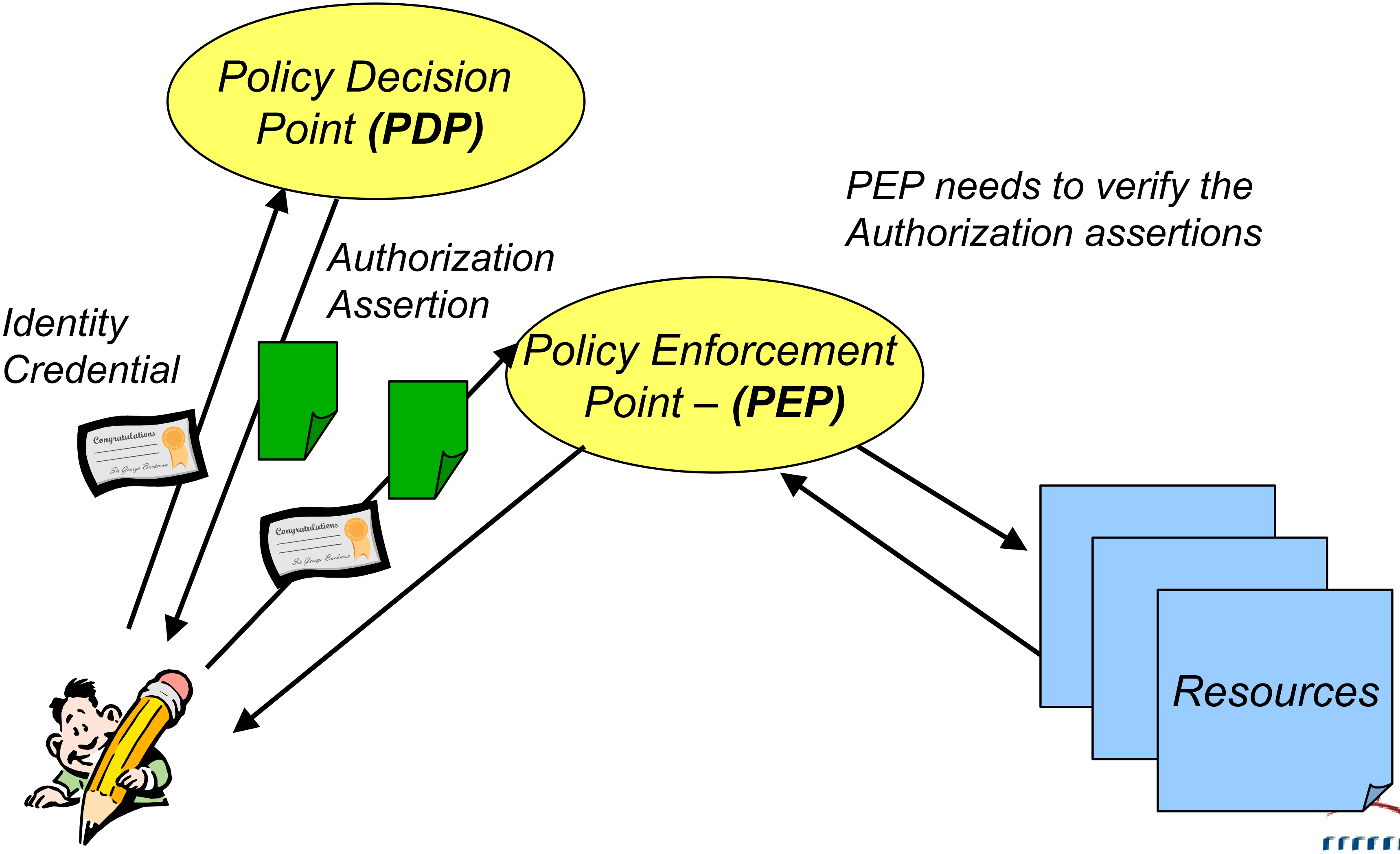
Authorization Decision

Policy Enforcement Point – ***(PEP)***

Identity Credential

Resources

Push Model



Policy Decision Point **(PDP)**

Identity Credential

Authorization Assertion

Policy Enforcement Point – **(PEP)**

PEP needs to verify the Authorization assertions

Resources

Recent Authorization Tokens

- **Digitally signed certificates**
  - Include the user’s name and are signed by a trusted authority
  - can be used to grant a user attributes –
    - IETF-draft An Internet Attribute Certificate Profile for Authorization, Farrell & Housley
  - Can be used to grant a user rights to a resource
    - CAS and Akenti
- **SAML assertions**
  - signed XML statements about rights.

Improvements Over Classic Capabilities

- Short life times make revocation less of a show stopper
- Digital signatures are viable way to make the capabilities non-forgeable
- The bearer of the capability must be able to prove that he is the user mentioned in the certificate. This eliminates unrestricted delegation of the rights

Delegation

- Delegation = remote creation of a (second level) proxy credential
    - New key pair generated remotely on server
    - Proxy cert and public key sent to client
    - Clients signs proxy cert and returns it
    - Server (usually) puts proxy in /tmp
- Allows remote process to authenticate on behalf of the user
    - Remote process "impersonates" the user

X.509 PKI Proxy Certificate Profile

- **Proxy certificate: signed for and derived from a “normal” X.509 End Entity certificate (or by another Proxy cert)**
- **Support for restricted delegation - new V3 extensions**
  - **Restricted rights**
  - **Tracing of delegation path**
- **Tricky subject: Are these identity certs or attribute certs?**
  - **Proxy authority is not Certificate Authority**
  - **Does not violate the EE signing restrictions contained in the X.509 keyCertSign field of the keyUsage extension.**

“Proxy Authority, Not CA”

- CA Roles
  - Naming
  - Binding name to public key
  - uses Registration Authority (RA)
- Proxy cert does not define new name
- EE Cert: subjects = name (name used for authorization)
- CA Cert: subject used for path validation (issues field in EE or CA *must* match subject name of CA cert)

Digital Signatures

- Combines a hash with an asymmetric encryption algorithm
- To sign
  - hash the data
  - encrypt the hash with the sender's private key
  - send data signer’s name and signature
- To verify
  - hash the data
  - decrypt the signature with the sender's public key
  - the result of which should match the hash

Elements of PMI
Privilege Management Infrastructure

- An effort within the IETF to define a structure similar to PKI for Privilege Management
- Attribute Authority instead of CA
- X.509 Attribute Certificates (an IETF standard)
    - Subject
    - Issuer
    - Attribute name and value
    - Validity dates
    - Signed by AA

GAA

- Generic Authorization and Access control APIs (GAA-API)
- Use GSSAPI to obtain principal's identity
- GAA-API gets policies from local files and distributed authorization servers
- Module provides for simple Extended Access Control Lists (EACLs)

GAA (cont.)

- Two main functions
  - gaa_get_object_policy_info
    - ACL-based: object ACLs; Capability-based: list of authorities allowed to grant capabilities
  - gaa_check_authorization
    - Yes, no, or “additional checks required”
- For more information
  - draft-ietf-cat-acc-cntrl-frmw-05.txt (expires Apr 2001)
  - draft-ietf-cat-gaa-cbind-05.txt
- Probably dead for now

Grid Authorization Services

- Globus grid-mapfile
  - Per site mapping of Grid Id (X.509 Cert) to local user id
  - Fine grained access control done by OS based on user id
- Akenti – certificate-based authorization policy
  - Allows for multiple stakeholders per resource
  - Distributed authorization policy
- Community Authorization Service (CAS)
  - Allows a site to give a block grant to a community of users (site can treat community as a group)
  - Fine grain access control is delegated to CAS server (to enforce community policy for individual users)
- VOMS –similar to CAS but run by VOs
  - Provides authorization tokens to members
  - Provides updates of Grid-map files to resource providers

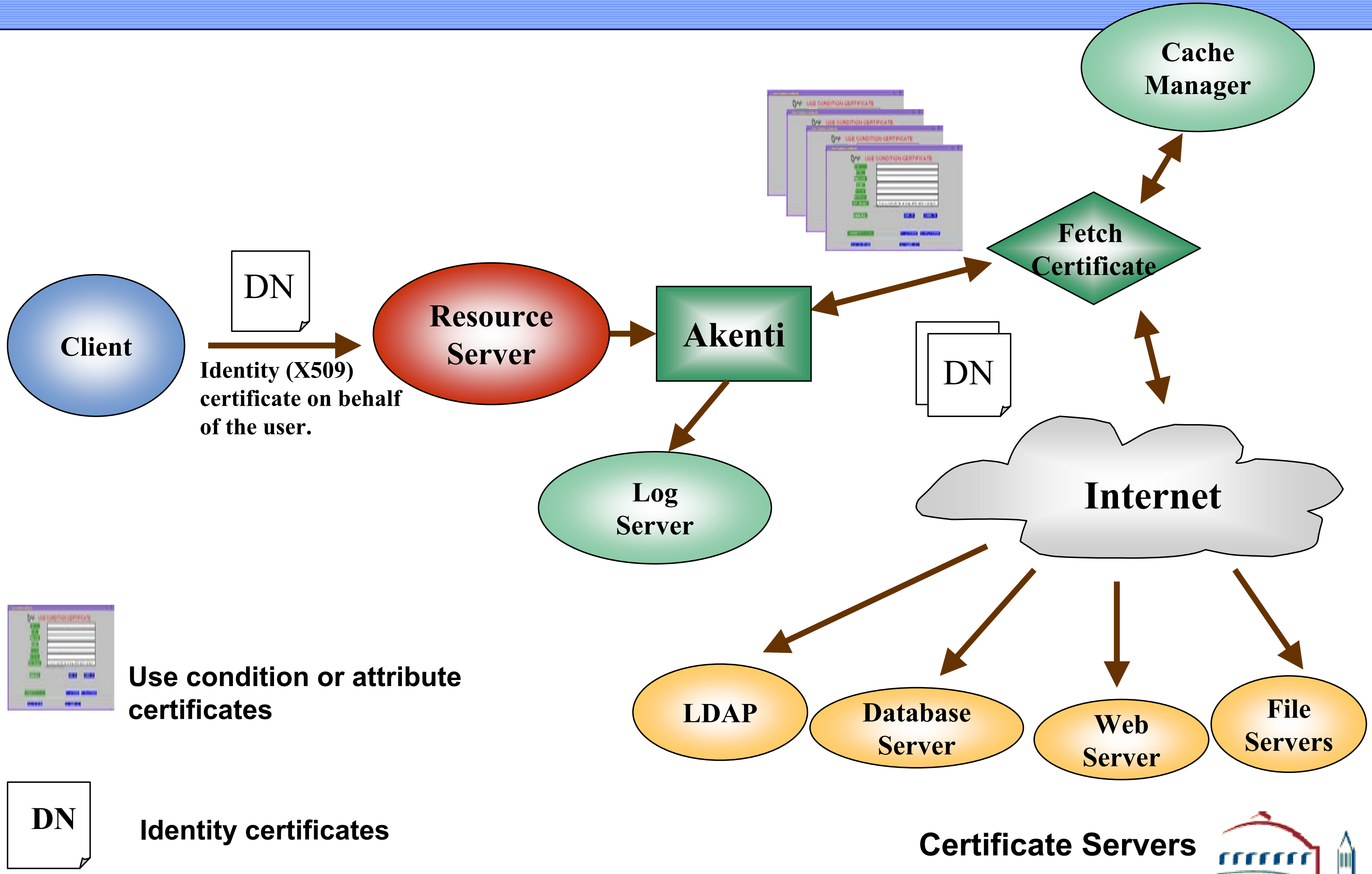
Akenti Goals

- Access based on policy statements made by stakeholders – no central administration
- Handle multiple independent stakeholders for a single resource
- Use Public Key Infrastructure standards to identify users and create digitally signed certificates
- Emphasize usability
  - Stakeholders can see all the policy for their resource
  - Access control decisions can be monitored

Akenti Access Control

- **Based on the following digitally signed certificates:**
  - **X.509 certificates for user authentication**
  - **UseCondition certificates containing stakeholder policy**
  - **Attribute certificates in which a trusted party attests that a user possesses some attribute, e.g. training, group membership**
- **All policy is kept in signed policy certificates.**
- **Minimal local Policy Certificate - Who to trust, where to look for certificates.**
- **Can be called from any application that has an authenticated user's identity certificate and a unique resource name, to return that users privileges with respect to the resource.**

Akenti Architecture



Cache Manager

Fetch Certificate

Client

DN

Identity (X509) certificate on behalf of the user.

Resource Server

Akenti

DN

Log Server

Internet

LDAP

Database Server

Web Server

File Servers

Use condition or attribute certificates

DN

Identity certificates

Certificate Servers

Required Infrastructure

- Certificate Authority to issue identity certificates (required)
- Method to check for revocation of identity certificates (required)
  - LDAP server – That identity certificates are published in
  - Certificate Revocation lists - supported by most CA's
  - Online Certificate Status Protocol (OCSP) - new standard
- Network accessible ways for stakeholders to store their certificates (optional)
  - Web servers
  - MSQL web accessible data bases
- SSL connections between users and resources

Status

- **Akenti was used by the Diesel Combustion Collaboratory to control**
  - secure data/image repository
  - ORNL electronic notebooks
  - PRE (CORBA) remote job executions
- **Is used by the National Fusion Collaboratory**
  - Control access to data archives (MDSPlus)
  - Fine-grain control of job executions started by the Globus Gatekeeper
- **Runs on Solaris and RedHat 7.2 Linux**

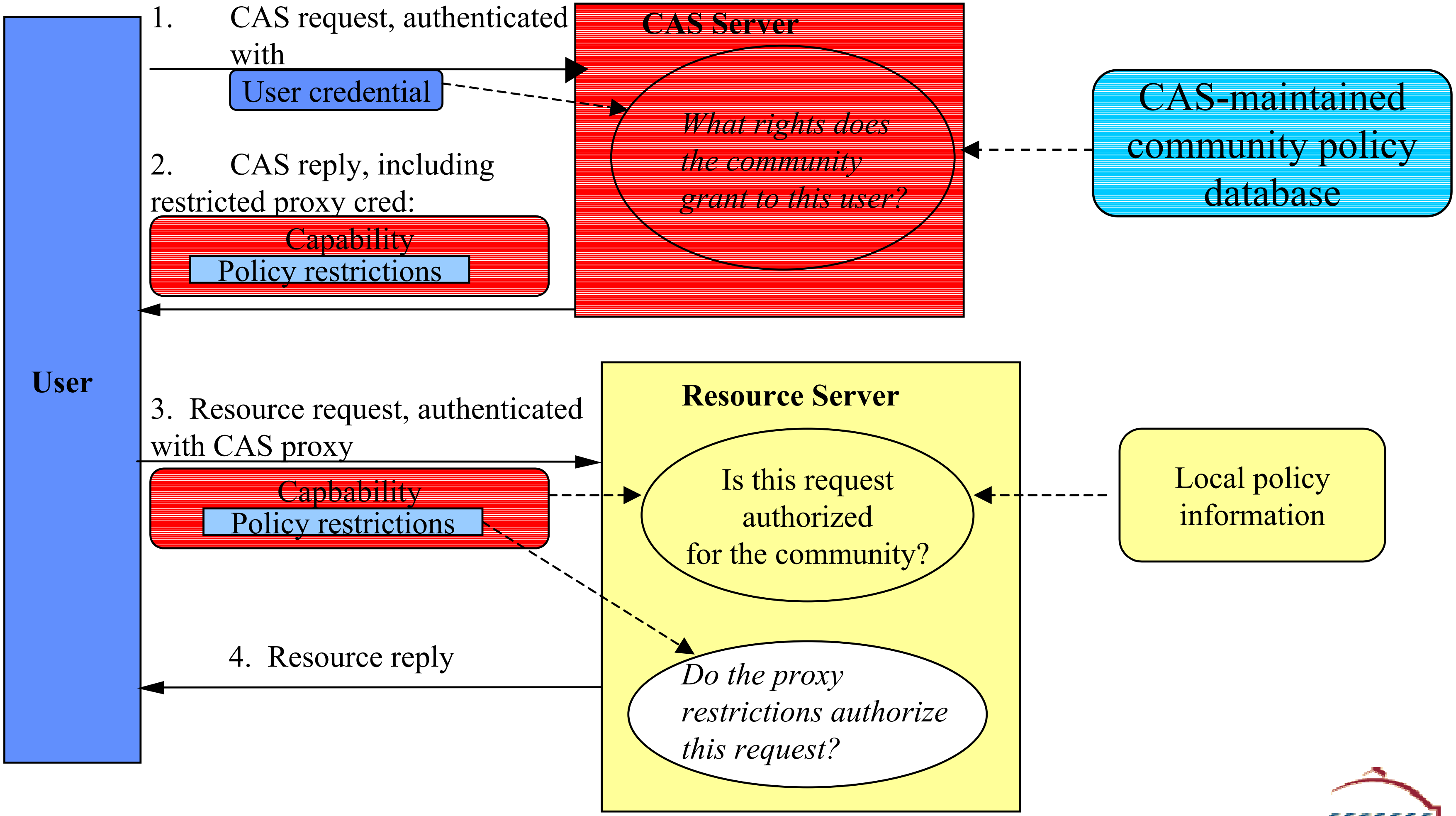
CAS Motivation and Goals

- Typically computational Grids are built to support the sharing of resources by a set of individuals or institutions
- These users of a Grid can be thought of as Virtual Organization or community of users.
- A resource provider may want to allow use of its resources to the community at large and let the community itself do the fine-grained access control of its users.
- Thus the resource provider does not need to know about all the members of a community
- Community members only have to deal with a community server and not many resource providers.
- CAS allows scalability since resource administrators do not need to deal with complexity of each VO

CAS Access Control

- The CAS server runs with its own identity
- That identity is granted broad access to all the resources in a Grid
- The user contacts the CAS server and gets a CAS restricted proxy containing the rights that he has which must be a subset of the CAS rights.
- CAS knows all the community members and their rights to all the Grid resources
- The user contacts the resource, presented the CAS proxy as its identity credential.
- The resource gatekeeper checks that CAS has the rights contained in the proxy and allows the access.

A Typical CAS Request



User

1. CAS request, authenticated with

User credential

CAS Server

What rights does the community grant to this user?

CAS-maintained community policy database

2. CAS reply, including restricted proxy cred:

Capability

Policy restrictions

Resource Server

3. Resource request, authenticated with CAS proxy

Capbability

Policy restrictions

Is this request authorized for the community?

Local policy information

Do the proxy restrictions authorize this request?

4. Resource reply

Slide courtesy of the Globus Project

Enabling Technologies and Infrastructure

- PKI – CA, Revocation mechanism
- GSI – support for proxy certificates
- Restricted Proxy Certificates IETF-draft
- CAS server
  - Interface to set policy
  - Authorization to set policy
- Libraries for resource gatekeeper to interpret the rights in a restricted proxy.

Status

- Started in 2001
- Early implementation demonstrated at HPDC Aug 2001 and SC01 Nov 2001
- Used by a version of GridGTP
- Released as part of GT2

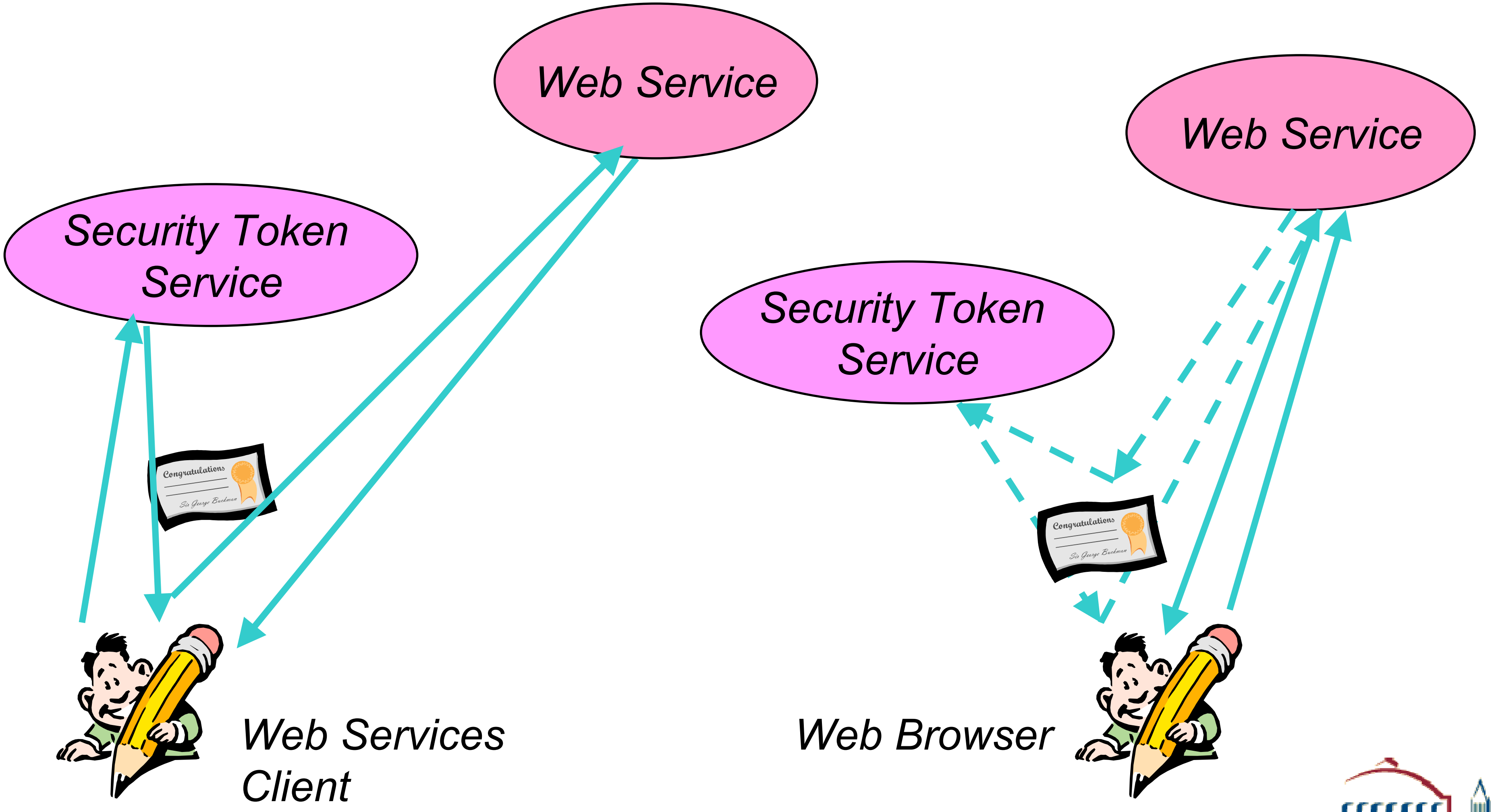
EDG VOMS

- VOMS server is similar to CAS but run by a Virtual Organization
  - It defines the membership and roles and groups of each VO member
- Integrated with Globus GT2
  - Modified grid-proxy-init to get a proxy from VOMS that includes the VO name and the user's groups and roles.
  - Modified the Gatekeeper to control admission on the basis of the VOMS proxy
  - The VOMS server also provides a grid-map-file derived from the VO members

Web Services

- Extend the Web browser/Web server model of loosely coupled interactions via self-describing messages.
- Expand clients beyond browsers
- Expand language from http to XML/SOAP
- Web services behaviors are described in yet another XML language called WSDL

Simple Web Service Authorization Model



Web Service

Security Token Service

Web Service

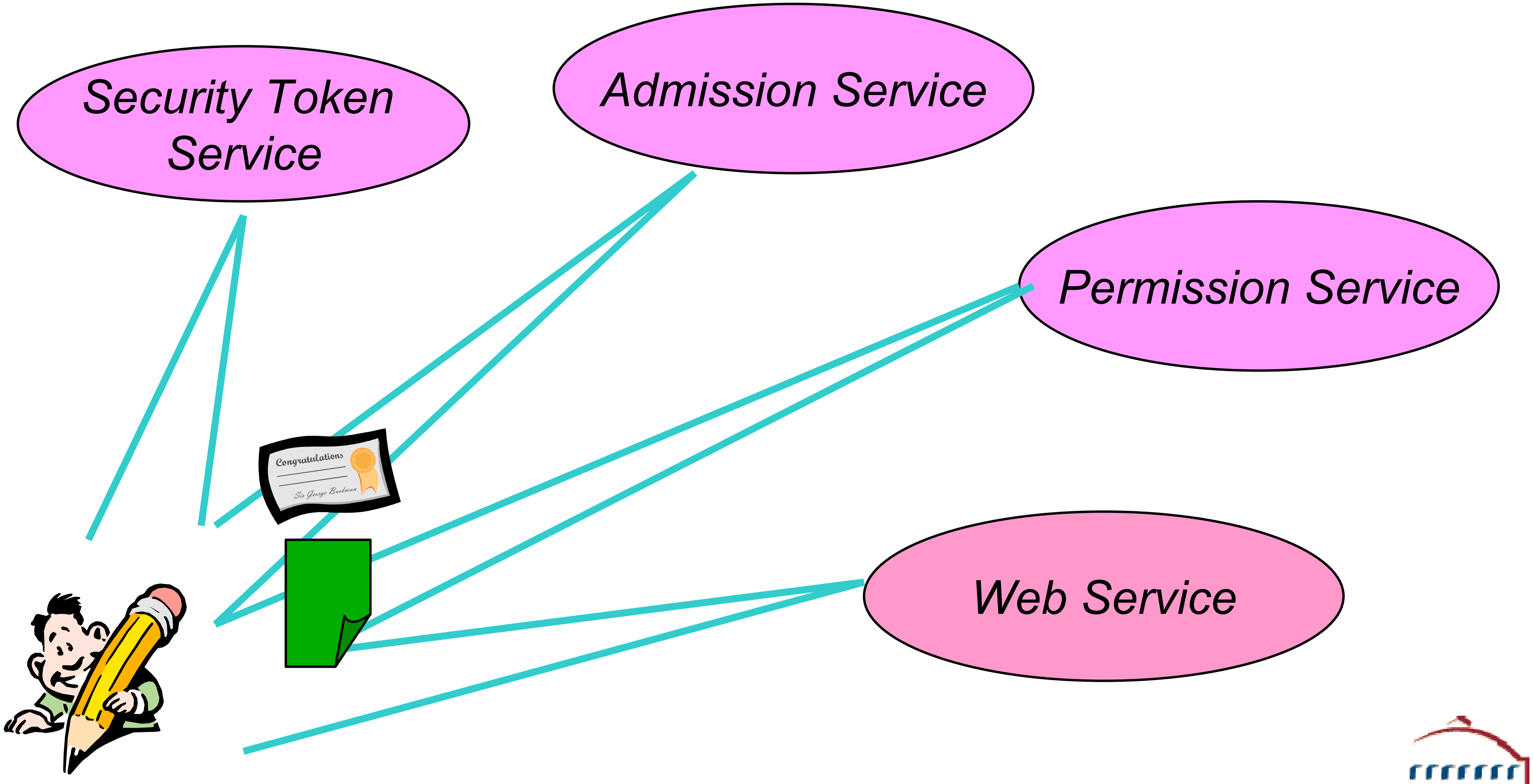
Security Token Service

Web Services Client

Web Browser

GXA – Global XML Web Services Architecture

Microsoft’s vision of Web security



Passport

- HUGE (200 million users, 90 sites) single-sign-on service in Redmond
    - Based on email address and password
    - May store a number of attributes about each user
- Also federated model
    - participating sites store their own data locally. May store PINs for it on Passport.
- May support additional levels of authentication and security based on Kerberos, smart cards, digital certificates

How Passport works

- User creates an account from any signon site
  - Passport Unique Identifier (PUID)
  - Email or phone number
  - Optionally name, zip code, gender …
  - Credential (password or pin, 1-3 secret questions)
- Single signon to all .NET Passport sites
  - Passport site redirects authorization to Redmond
  - Redmond asks for and checks credentials
  - Creates a cookie encrypted with requesting site's public key – includes PUID and other information
  - Cookie get forwarded to Passport site who keeps your profile under your PUID.

Liberty Alliance

- **Sun, HP, MasterCard, AMEx, Entrust, AOL, Internet2**
- **Goal - to develop and deploy open, <u>federated</u> solution for network identity**
  - **Allow individuals and business to keep personal information securely**
  - **Provide a universal, open standard for "single sign-on"**
  - **To provide an open standard for network identity**
- **"Circle of Trust"**
- **Formed Sept 2001**
- **Protocol specifications based on SAML and SOAP bindings - V1.0 Jul 02, V1.1 rfc Jan 03**

Liberty Alliance V1.1 Specs

- Opt-in account linking
- Simplified sign-on for linked accounts
- Authentication context
  - "Institutions or companies linking accounts can communicate the type and level of authentication that should be used when the user logs into different accounts."
- Global log-out

Liberty Alliance v1.1 specs

- **Liberty Architecture Overview** is a non-normative architectural overview to Liberty that describes the protocols and offers policy and security guidance.
- **Liberty Protocols and Schema Specification** defines the abstract protocols and XML schemas for Liberty.
- **Liberty Bindings and Profiles Specification** defines concrete transport bindings and usage profiles for the abstract Liberty protocols.
- **Liberty Authentication Context Specification** defines the authentication context schema, which is used to communicate information about an authentication event.
- **Liberty Glossary** contains Liberty terminology.
- **Liberty Architecture Implementation Guidelines** provides non-normative implementation and deployment guidance for the various entities in a Liberty-enabled network.

Shibboleth and Anonymous Credentials

- Part of the Internet2 middleware project
- User authenticates at home site
- References another co-operating site which can ask the originating site for a handle for the user
- Asks server at originating site for attributes about the handle which it can use for authorization
  - E.g. user is a faculty member.
- Currently implemented with modified browsers and web servers.

XML Authorization/Privilege languages

- **SAML – authorization requests**
- **XACML – access control policy**
- **XrML – digital rights expression**
- **WS-Security – SOAP extension**
- **WS-Security Policy – SOAP extension**

SAML

- Organization for the Advancement of Structured Information Standards (OASIS)
  - Develops industry standards specifications for interoperability based on XML
- SAML: Security Assertion Markup Language
  - Derived from AuthXML and S2ML
- A framework for exchanging authentication and authorization information
  - Support single-sign-on for Web Services
  - Assertions – about authentication, authorization and attributes
  - Protocol – request and respond messages

XACML

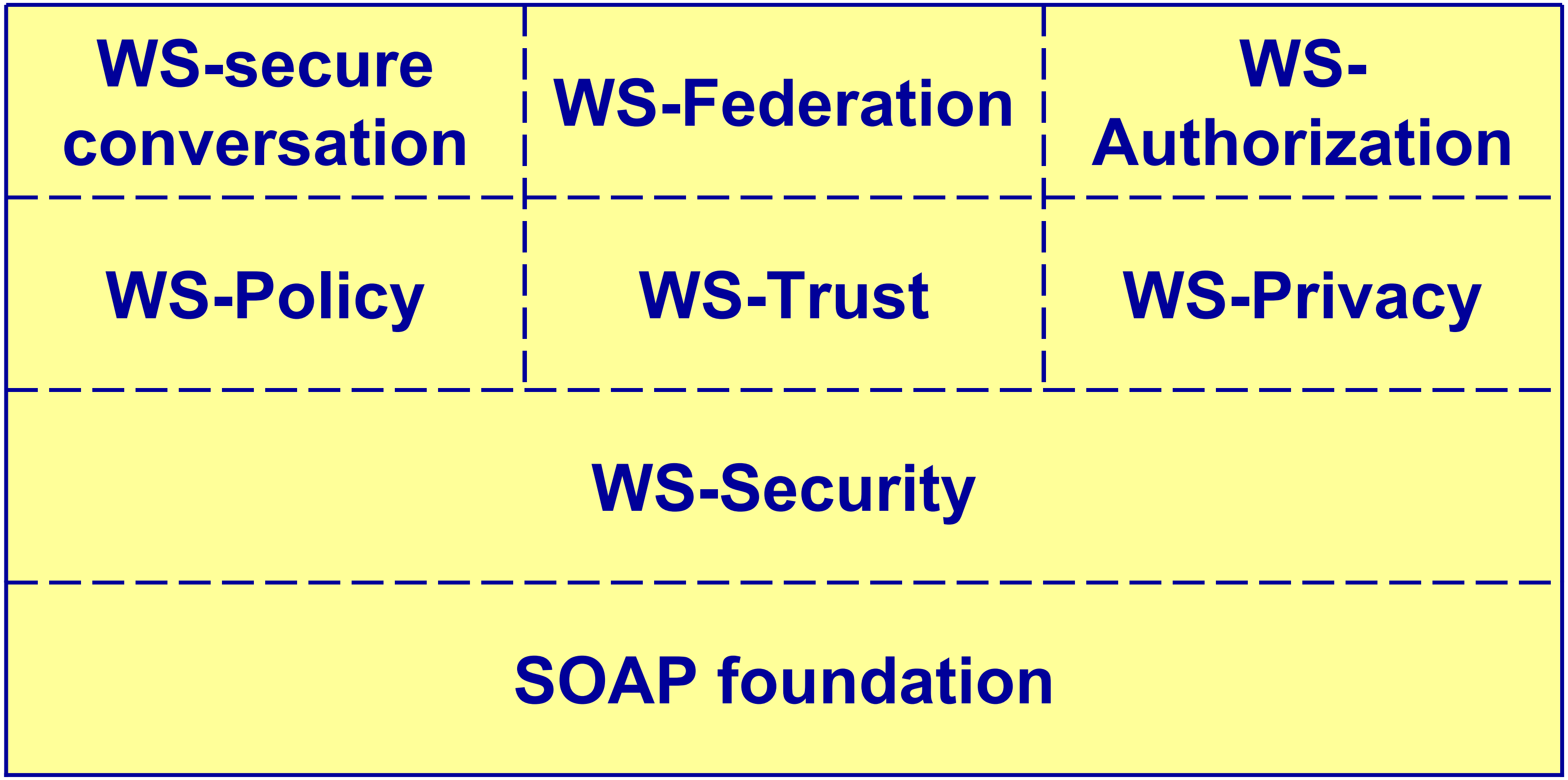
- Describes authorization policy
  - policies
    - rules, rule combining algorithm, obligations
  - rule contains
    - target , effect, condition
  - target
    - subject, resource, action
- Describes the security context of the user
  - Request
    - subject, action desired, resource, environment (attributes)
  - Response
    - resource id, decision, obligations, status

XrML

- Specifies rights and conditions to control the access to digital content and services
- From the digital media industry
- Defines a license (right to use)
- Not adapted by a standards body yet, but has been submitted to OASIS rights language technical committee

Web Services Security Specifications

A Joint white paper from IBM, Microsoft


WS-secure conversation

WS-Federation

WS-Authorization

WS-Policy

WS-Trust

WS-Privacy

WS-Security

SOAP foundation

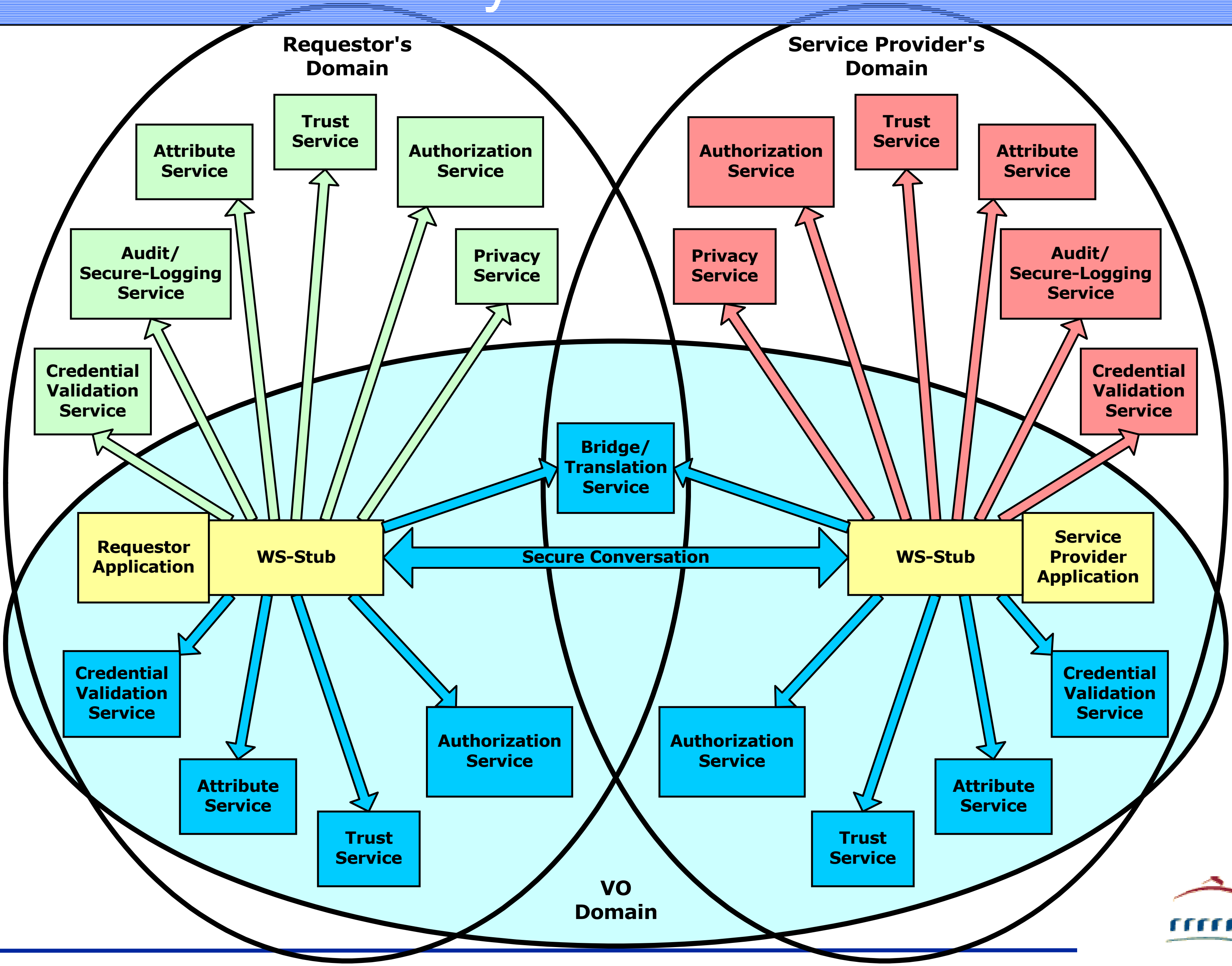
WS-Security Policy

- Security Token Assertion
  - Type
    - UsernameToken – username and password
    - BinarySecurityToken – X509, Kerberos TGT or ST
    - X509v3
    - SAMLAssertion
    - XrMLLicense
  - Issuer
  - Claims
- Integrity Assertion
- Confidentiality Assertion
- Age of message

WS-Trust

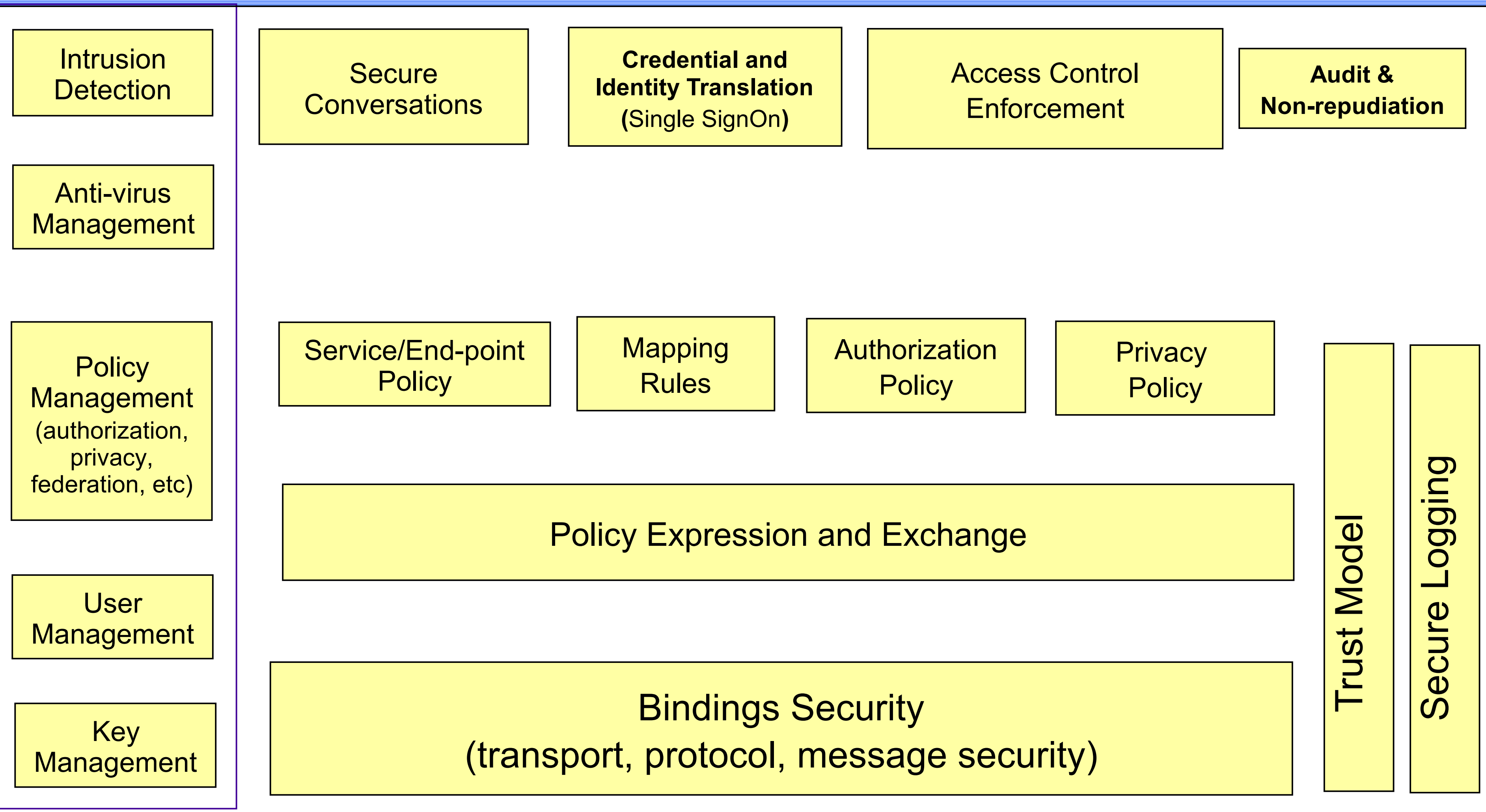
- Methods for issuing and exchanging security tokens
- Ways to establish and access trust relationships
- RequestSecurityToken
  - Key and encryption requirements
  - Delegations, forwarding and proxy requirements
  - Lifetime and renewal requirements
  - Policies
- RequestSecurityTokenResponse
  - SignChallenge (optional)

Security Services & VO



From *OGSA Security Roadmap* Frank Siebenlist, et.al.

OGSA Security Components



Intrusion Detection

Anti-virus Management

Policy Management (authorization, privacy, federation, etc)

User Management

Key Management

Secure Conversations

Credential and Identity Translation (Single SignOn)

Access Control Enforcement

Audit & Non-repudiation

Service/End-point Policy

Mapping Rules

Authorization Policy

Privacy Policy

Policy Expression and Exchange

Bindings Security
(transport, protocol, message security)

Trust Model

Secure Logging

From *Security Architecture for Open Grid Services* Nataraj Nagaratnam, et. al.

Building Blocks

Exploiters	AppServer security	Platform (OS) security	Application security (on top of app server)			
Security services *(TBD)*	AuthnService	AttributeService	AuthzService	. . .	AuditService	
Federation layer	WS-Federation	WS-SecureConversation	WS-Authorization			
Policy layer	WS-Policy	WS-Trust	WS-Privacy			
Message Security	ds: Signature	xenc: EncryptedData	. . .	SecurityToken		
Web services standards	WSDL	SOAP	. . .	WS-Routing	WS*L	
XML security standards	XML Signature	XML Encryption	Assertion Language	XKMS	. . .	*XACML*
Protocol layer security	HTTP https	IIOP CSIv2	. . .	JMS (MQ)		
Platform resource security	NT	Solaris	Linux	AIX	OS/400	z/OS

From *Security Architecture for Open Grid Services* Nataraj Nagaratnam, et al.

Use of SAML for OGSA authorization

- Defines the SAML constructs to use in authorization messages between
  - The requestor and an authorization agent (getting authorization assertions)
  - The requester and the PEP (pushing authorization assertions
  - The PEP and PDP (requesting and returning authorization assertions)
- Policy may be written in terms of operations and Service Data Element (SDE) on a service

Elements of Authorization Request

- **Subjects defined by NameIdentifier**
- **Resources defined by URI**
  - Grid Service Handle (GSH) for services
  - Wildcards defined (rights on all the resources a PDP controls)
- **Actions**
  - Maybe be an operation on resource (service)
  - Read or modify of a SDE
  - Wildcards defined
- **Evidence**
  - Additional assertions on request

Elements of Authorization Decision

- **Conditions**
  - specifying additional conditions for use
- **Advice**
  - Can be used to provide proof of assertion
  - May be ignored
- **Authorization Decision element**
  - Same items as in the AuthorizationQuery
- **Signature**
  - Assertions should be signed if they were not provided over a secure channel.

SAML vs XACML

- **SAML**
  - More mature
  - Simpler – designed for Authorization queries and assertions
  - Responses contain Authorization Assertions
    - Subject, resource, actions, conditions, evidence, advice and signature
- **XACML**
  - Designed to express policy as well as answer requests about authorization
  - Responses are just an answer
    - ResourceID, Permit/Deny/Indeterminate, Obligations

Current GGF Security Area

- OGSA-Security Working Group
  - Document on “Use of SAML for OGSA Authorization”
- Authorization Frameworks and Mechanism
  - Glossary
  - Grid Authentication Requirements
  - Conceptual Grid Authorization Framework and Classification
- Site Authentication, Authorization and Accounting Requirements Research Group
  - The purpose of this research group is to collect and codify the requirements of existing grid resource sites with respect to the acceptance of grid credentials for access to their services.